

CS106 W2020 - Lab 04

Due: Wednesday, January 29, 2020, 11:59 PM

This lab is about Advanced Shapes.

Starter Code is provided. The project name for the start code is: "sketch_starter_code".

Question 1: Singles

*This question contains additional function-writing exercises, similar to the "Singles" question from Lab 01. It will help develop your ability to write small functions in larger programming assignments. However, **you do not need to submit your code**. Complete the functions and continue with the rest of the lab.*

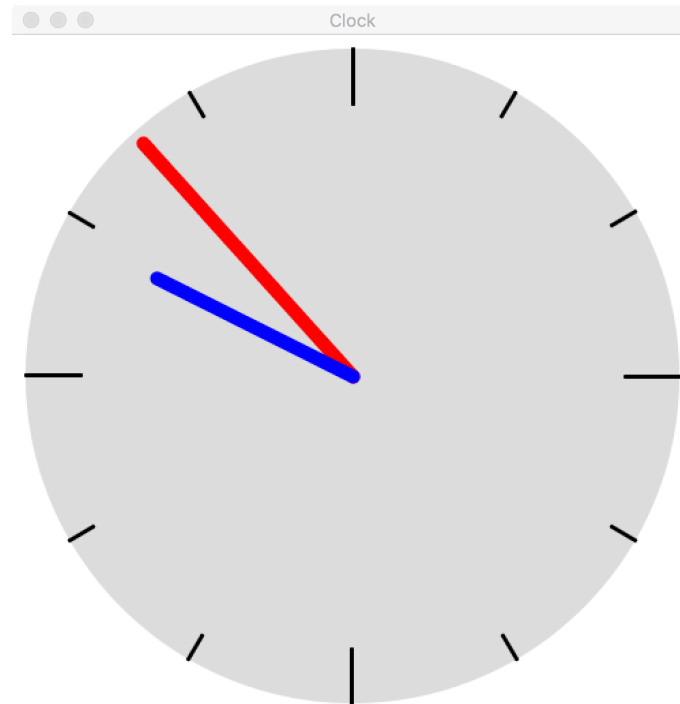
In your L04 folder, create a new sketch called Singles. In that sketch, write the following functions.

1. Write a function called `circle()` that takes three floating-point numbers as arguments, representing the x and y coordinates of the centre of a circle and a radius for that circle, and draws the circle. The function should not return a value.
2. As an extension, write a second version of `circleVec()` that takes two arguments: a single `p5.Vector` containing the centre of the circle, and a radius.
3. Write a function `secondsToday()` that takes no arguments. It uses the built-in functions `hour()`, `minute()` and `second()` to return an integer telling you how many seconds have elapsed since midnight. So, if the function was called at exactly 2:37.19pm, it would return $14 * 60 * 60 + 37 * 60 + 19 = 52639$.
4. Write a function `fourCorners()`. It takes no arguments, and returns an array of four `p5.Vector` objects, representing the four corners of the sketch window. For example, if `setup()` contains the call `createCanvas(400, 300);`, then `fourCorners()` might return the array `[createVector(0, 0), createVector(400, 0), createVector(400, 300), createVector(0, 300)]`.

You are encouraged, but not required, to test your functions by writing a `setup()` function that calls them and checks whether the correct answers are returned.

If you want, save your work in the sketch Singles.

Question 2: Clock



Create a working analogue clock sketch that displays the current time. The sketch window should have size at least 500×500. It should show 12 tick marks around the clock face, with the marks for 12, 3, 6 and 9 o'clock drawn more prominently (e.g. longer). There should be a minute hand and a shorter hour hand that show the current time. The hour hand should be offset by the number of minutes after the hour, so that it sweeps smoothly from one hour to the next.

The starter code for this lab draws the background circle and tick marks on the clock. You can copy the relevant code from “sketch_starter_code” and modify as per your requirements.

Here's a suggested plan for writing this sketch:

1. In your L04 folder, create a new sketch called Clock.
2. Write a `setup()` function that sets the sketch window size to (at least) 500×500.
3. Copy the function `polar()` from the lecture notes into the sketch. You'll want to use it here.
4. Create a `draw()` function. Set the background colour. If you want, draw a background for the clock face (e.g., a circle).
5. In `draw()`, write a for loop that draws 12 tick marks for the hours. You can use any shapes you want, though lines or triangles are probably simplest. The marks for 12, 3, 6

and 9 o'clock must be visually distinct. Use the `polar()` function to calculate the coordinates of the points you want to draw. (If you want, you can try drawing even smaller tick marks for the minutes too.) Remember that your coordinates must be offset relative to the centre of the sketch.

6. Use the built-in functions `hour()` and `minute()` to ask P5.js for the current time. Store these values in local variables. Look at the documentation for these functions if you need to.
7. Draw the minute hand as a line from the centre of the clock to the edge. Use the current minutes after the hour to determine what angle to use with the `polar()` function. Remember that the sketch's coordinate system is "upside-down", so you may need to fiddle with angles a bit, and/or turn some numbers into their negatives.

Hint: Built-in `map()` function might be useful

8. Draw the hour hand as a line from the centre of the clock to a point not quite at the edge. Use the current hour to determine the angle, but also add in a fractional amount of hour based on the current minutes. For example, at 10:30, the hour hand should point halfway between 10 and 11; at 7:23, the hour hand should point to a position 23/60 of the way from 7 to 8.

Hint: Built-in `map()` function might be useful

Save your work in a sketch titled Clock

Submission

When you are ready to submit, please follow these steps.

1. Please ensure that any sketches you submit compile and run. It's better to submit a sketch that runs smoothly but implements fewer required features than one that has broken code for all features. If you get partway into a feature but can't make it work, comment it out so that the sketch works correctly without it.
2. If necessary, review the Code Style Guide and use Processing's built-in auto format tool. You do not need to use the precise coding style outlined in the guide, but whatever style you use, your code must be clear, concise, consistent, and commented.
3. If necessary, review the How To Submit document for a reminder on how to submit to LEARN.
4. Make sure to include a comment at the top of all source files containing your name and student ID number.
5. Create a zip file called L04.zip containing the entire L04 folder and its subfolders. If your folder also contains your Singles sketch, that's fine.

6. Upload L04.zip to LEARN. Remember that you can (and should!) submit as many times as you like. That way, if there's a catastrophe, you and the course staff will still have access to a recent version of your code.